Testing Application Behavior on Frequency Over-scaled Micro-controllers

Jingjing Zhou
University of Minnesota, UROP Project
Advised by Prof. John Sartori

Introduction

For several emerging applications such as wearables, internet of things, and sensor networks, energy efficiency is of great importance. General purpose embedded processors (microcontrollers) are the preferred solution for many such applications due to the evolving nature of these applications and the high costs of custom IC design.

This project will explore a phenomenon called dynamic timing slack that may enable important improvement of energy efficiency in micro controllers. Dynamic timing slack exists when the embedded software application executed on a processor does not exercise the processor's static critical paths. Static critical paths are the logic paths in the processor that have the longest delays. In such scenarios, the longest path that is exercised by the application (le us call it the dynamic critical path) has additional timing slack that can be exploited for performance improvement or power savings by scaling up the operating frequency or scaling down the voltage of the processor. Interestingly, non-exercised paths can be allowed to violate timing constraints (i.e., have delay that is longer than the clock period) while still maintaining error-free application execution. This is because a non-exercised path can never cause an error.

Reference

1. John Sartori and Rakesh Kumar. "Exploiting Workload-dependent Timing Slack for Energy Efficiency in Embedded Systems". 51st ACM/IEEE DAC, June 2014.

Approach

The method of showing dynamic timing slack is to test a collection of applications on several microcontrollers operating at over-scaled frequencies and determine if any of the applications can execute correctly at frequencies that are higher than the rated frequency of the microcontroller.

- Collecting microcontrollers: build different circuits on the breadboard that can test different benchmark applications of different microcontrollers
- Collecting some benchmark applications: write codes with different functions that can be applied to different microcontrollers
- Assemble test framework
- Execute applications and record results
- Debug applications to determine causes of failure: run snippets of applications

Testing Results

The microcontrollers that tested in this projects are PIC16F916, PIC18F4550, PIC16F88, PIC24FJGA002, and MSP430. The results are as below:

Result of PIC16F916, rated frequency = 20MHz

number	description	max frequency the micro-controller can work
1	basic application of I/O port	33.75MHz
2	basic operation	33.25MHz
3	more complex operation	22.325MHz
4	PWM	21.775MHz
5	LCD application	14.125MHz
6	LCD application and timer	13.425MHz

Result of PIC16F88, rated frequency = 20MHz

number	description	max frequency the micro-controller can work (trun off the trun on) at 3.03V
1	multiply	42.25MHz
2	division	40MHz
3	Avergae	41.5MHz
4	thold	42.75MHz
5	insort	42.5MHz
6	intfilt	41.0MHz
7	autocorr	43.75MHz
8	convolution	40.5MHz

Testing Results Continue

Result of PIC18F4550, rated frequency = 48MHz

number	description	max frequency the micro-controller can work (trun off the trun on)
1	simple operation	57.3MHz
2	random number	54.735MHz
3	LED blink by fixed delay time	56.63MHz
4	timer	55.68MHz
5	comparator	55.955MHz
6	PWM	55.15MHz
7	adc	55.00MHz

Result of PIC24FJGA002, rated frequency = 32MHz

number	description	max frequency work at 3.3V	voltage range at 32MHz
1	simple operation	45.245MHz	1.75v-3.6V
2	more complex operation	42.685MHz	1.78V-3.6V
3	timer (poilling)	57.8MHz	1.75V-3.6V
4	timer (interrupt)	42.5MHz	1.77V-3.6V
5	outout compare (mode1)	45.82MHz	1.75V-3.6V
6	output compare (PWM)	46.35MHz	1.75V-3.6V
7	ADC and SPI	42.9MHz	1.80V-3.6V
8	ADC and UART	43.2MHz	1.77V-3.6V

Result of MSP430, rated frequency = 8MHz

number	description	max frequency the micro-controller can work when the supply voltage is 3.23V	maxmium frequency at supply voltage is 3.6V
1	mutiply with blinking	14.2MHz	14.5MHz
2	division with blinking	13.1MHz	14.5MHz
3	average with blinking	13.1MHZ	14.5MHz
4	rle with blinking	13.2MHz	14.5MHz
5	tHold with blinking	13.2MHz	14.5MHz
6	insort with blinking	13.0MHz	14.5MHz
7	binsearch with blinking	13.0MHz	14.6MHz
8	intfilt with blinking	12.9MHz	14.5MHz
9	tea8 with blinking	12.9MHz	14.5MHz
10	FFT with blinking	12.3MHz	13.9MHz
11	autocorr with blinking	13.1MHz	14.6MHz
12	convolution with blinking	13.1MHz	14.6MHz

Conclusions

All the chips I tested can be over scaled 15% to 100% beyond the rated clock frequency for different microcontrollers, which means that their performance and efficiency still have large room to be improved. This results also used in a paper that have been accepted by ISCA.